Period: 1
Name: Andrew Park
Group Name: Diseased
Project Title: Disease Simulator

Project Description: This project aims to simulate the spread of a disease and how it affects a population in a user-inputted environment. Some variables include the disease's deadliness, its spread rate (k-factor) and the number of persons.

List of Current Functionalities:
1. First, the program will generate multiple successive windows that ask for inputs in the following categories: the number of persons, people's starting health, the infection spread rate and the infection deadliness rate. This helps set up the parameters for the environment and how the disease is configured. We make this user-definable because we want to see how different environments and different diseases change infection rates and times.
2. Next, the environment is initialized with one infected person (Patient Zero). Then, the function draw is a continuously repeating function with a delay of 20 milliseconds (So everything isn't instantaneous), updating the environment, letting all the people move randomly and spreading the disease.
3. On the upper right hand corner, there is a small visual interface that says how many people are alive, how many people are infected, and how much time has passed (in seconds).

Libraries Used:
I used the library *javax.swing.JOptionPane* to allow for the user input windows. This lets me generate multiple windows before the simulation that lets the user configure the parameters of the environment and the disease. This is crucial, because the point of my project is to show how different diseases can change how environments are affected.
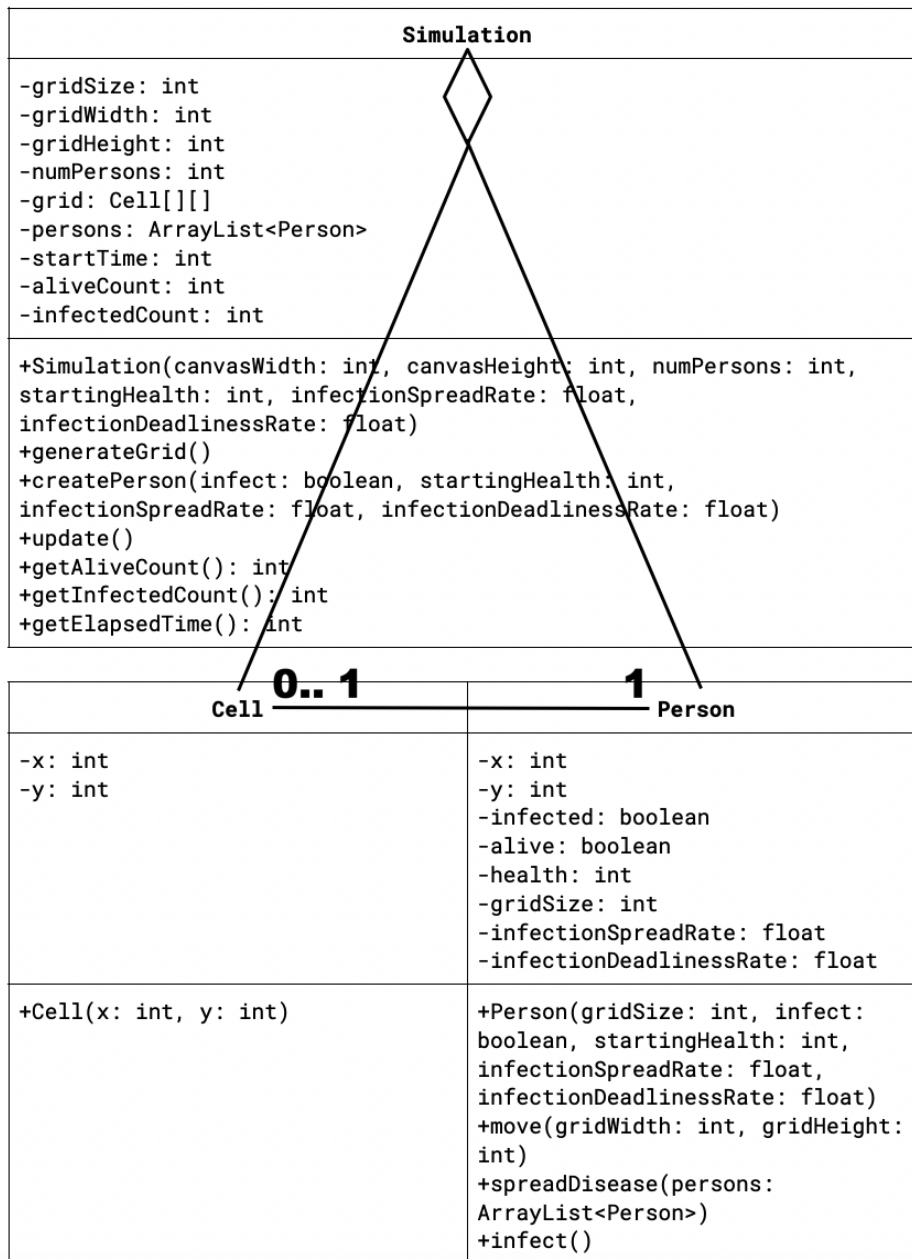
Extra:
I was inspired by this because I remember during the beginning of the COVID-19 pandemic, people were trying to model how the disease would

spread! This Processing project is essentially a simplified version that tries to emulate disease spread too, except we don't have data for specific symptoms and global population maps. I thought this was an interesting way to learn about diseases while also implementing real-world ideas in code.

UML Diagram:
There is association between Person and Cell.
There is aggregation from Simulation to Cell and Person.

Simulation

```
-gridSize: int
-gridWidth: int
-gridHeight: int
-numPersons: int
-grid: Cell[][]
-persons: ArrayList<Person>
-startTime: int
-aliveCount: int
-infectedCount: int
```
```
+Simulation(canvasWidth: int, canvasHeight: int, numPersons: int,
startingHealth: int, infectionSpreadRate: float,
infectionDeadlinessRate: float)
+generateGrid()
+createPerson(infect: boolean, startingHealth: int,
infectionSpreadRate: float, infectionDeadlinessRate: float)
+update()
+getAliveCount(): int
+getInfectedCount(): int
+getElapsedTime(): int
```

0.. 1 ——— **1**

Cell

```
-x: int
-y: int
```
```
+Cell(x: int, y: int)
```

Person

```
-x: int
-y: int
-infected: boolean
-alive: boolean
-health: int
-gridSize: int
-infectionSpreadRate: float
-infectionDeadlinessRate: float
```
```
+Person(gridSize: int, infect:
boolean, startingHealth: int,
infectionSpreadRate: float,
infectionDeadlinessRate: float)
+move(gridWidth: int, gridHeight:
int)
+spreadDisease(persons:
ArrayList<Person>)
+infect()
```

How does it work:
Once you press the run button, there will be windows that pop up one after another, asking you to input parameters of the environment as well as the disease. Fill them out according to the instructions, but also based on how you want the simulation to be configured. Once you've filled out all the windows, the simulation will run as you configured it.